

Deterministic Parikh automata on infinite words

Mario Grobler
grobler@uni-bremen.de
University of Bremen
Bremen, Germany

Sebastian Siebertz
siebertz@uni-bremen.de
University of Bremen
Bremen, Germany

ABSTRACT

Various variants of Parikh automata on infinite words have recently been introduced and studied in the literature. However, with some exceptions only their non-deterministic versions have been studied. In this paper we study the deterministic versions of all variants of Parikh automata on infinite words that have not yet been studied in the literature. We compare the expressiveness of the deterministic models, study their closure properties and decision problems with applications to model checking. The model of deterministic limit Parikh automata turns out to be most interesting, as it is the only deterministic Parikh model generalizing the ω -regular languages, the only deterministic Parikh model closed under the Boolean operations and the only deterministic Parikh model for which all common decision problems are decidable.

CCS CONCEPTS

• Theory of computation → Automata over infinite objects.

KEYWORDS

Parikh automata, infinite words, determinism, model checking

ACKNOWLEDGMENTS

We thank Georg Zetsche for insightful discussions and valuable remarks.

1 INTRODUCTION

Finite automata operating on infinite words find many applications in the world of formal languages, logic, formal verification, games, and many more. In many scenarios, non-determinism adds expressiveness or succinctness to their deterministic counterparts. However, this often comes at the price of important decision problems becoming hard to solve, or even undecidable [4, 23].

This is not only true for the most common model of finite automata operating on infinite words, namely Büchi automata (which recognize the ω -regular languages), but also for various extensions leaving the ω -regular realm. These include Parikh automata (PA) on infinite words, which gained a lot of attention just recently [11, 12]. Such automata are equipped with a constant number of counters (positive integers), and the counter values can be checked to satisfy some linear constraints. Several acceptance conditions for Parikh automata on infinite words have been studied in the literature, including deterministic and non-deterministic reachability PA, safety PA, Büchi PA and co-Büchi PA [12], as well as non-deterministic limit PA, reachability-regular PA, and strong and weak reset PA [11]. For all of these models, except safety PA and co-Büchi PA, testing emptiness is coNP-complete and universality

is undecidable [11, 12]. In contrast, for safety PA and co-Büchi PA, both problems are undecidable; however, requiring determinism yields a coNP-complete universality problem.

This motivates the study of the deterministic variants of the remaining models, namely deterministic limit PA, reachability-regular PA, strong reset PA, and weak reset PA. In this paper we study their expressiveness, closure properties and common decision problems. Grobler et al. [11] have shown that almost all non-deterministic variants of the aforementioned models (the exception being safety PA and co-Büchi PA again) form a hierarchy:

$$\begin{aligned} \text{reach PA} &\subseteq \text{reachability-regular PA} = \text{limit PA} \\ &\subseteq \text{Büchi PA} \subseteq \text{weak reset PA} = \text{strong reset PA}. \end{aligned}$$

First, we show that this is not the case for their deterministic variants. While

deterministic strong reset PA \subseteq deterministic weak reset PA
and

$$\begin{aligned} &\text{deterministic reachability PA} \\ &\subseteq \text{deterministic reachability-regular PA} \\ &\subseteq \text{deterministic weak reset PA} \end{aligned}$$

still holds, all other models become pairwise incomparable. Furthermore, we show that among all studied deterministic models only deterministic limit PA generalize Büchi automata in the sense that they recognize all ω -regular languages.

Deterministic limit PA also shine in the light of closure properties: among all studied PA operating on infinite words (the deterministic variants as well as the non-deterministic ones) they are the only ones closed under union, intersection and complement. This benefit also yields decidable decision problems. In contrast to the other models that were studied in [11, 12], emptiness and universality are decidable for deterministic limit PA. We show that also strong reset PA benefit from determinism: although having bad closure properties their universality problems becomes decidable. However, as we show, for deterministic reachability-regular PA and deterministic weak reset PA the universality problem remains undecidable.

	U	∩	¬
deterministic limit PA	✓	✓	✓
deterministic reachability-regular PA	✗	✗	✗
deterministic weak reset PA	✗	✗	✗
deterministic strong reset PA	✗	✗	✗

Table 1: Closure properties.

	$L = \emptyset?$	$uv^\omega \in L?$	$L = \Sigma^\omega?$
det. limit PA	coNP-c.	NP-c.	decidable
det. reach-reg. PA	coNP-c.	NP-c.	undecidable
det. weak reset PA	coNP-c.	NP-c.	undecidable
det. strong reset PA	coNP-c.	NP-c.	decidable

Table 2: Decision problems.

Finally, we study their inclusion and intersection emptiness problems, as these are important problems in order to study model checking problems. Here, we are given a system \mathcal{K} as a Kripke structure (a safety automaton) or a PA and a specification \mathcal{A} as a PA. The question whether at least one computation of a Kripke structure satisfies the specification (which we call existential safety model checking) has been studied in [11]. The authors show that this problem is coNP-complete for (nondeterministic) reset PA and for all models that are weaker. Similarly, the question whether all computations of a PA satisfy the specification (which we call universal PA model checking) has been studied in [12]. The authors show that this problem is coNP-complete for deterministic safety and co-Büchi PA, and undecidable for their nondeterministic counterparts as well as for deterministic reachability and Büchi PA. We study this problem as well as the universal safety model checking problem and the existential PA model checking problem for the remaining deterministic models. Again, deterministic limit PA shine as all these problems are decidable for them. For the other models, some problems are decidable and some are not.

In the following table we list our results. Some of the questions have already been studied in the literature or follow immediately from the results of [11, 12]. We fill the missing gaps and remark that these results are the technically most demanding results of this paper.

	Model Checking			
	Kripke		PA	
	\exists	\forall	\exists	\forall
det. limit PA	✓ [11]	✓	✓	✓
det. reach-reg. PA	✓ [11]	✗	✓	✗
det. weak reset PA	✓ [11]	✗	✗	✗
det. strong reset PA	✓ [11]	✓	✗	✓
det. reachability PA	✓ [11]	✗ [12]	✓ [12]	✗ [12]
det. Büchi PA	✓ [11]	✗ [12]	✓	✗ [12]
det. safety PA	✗ [12]	✓ [12]	✗ [12]	✓ [12]
det. co-Büchi PA	✗ [12]	✓ [12]	✗ [12]	✓ [12]

Table 3: Decidability of the model checking problem.

1.1 Related work

Parikh automata on finite words were introduced by Klaedtke and Ruess in [18] in order to show that an extension of weak monadic second-order logic (MSO) with cardinality constraints is decidable. PA turn out to be equivalent to Ibarra's reversal bounded multi-counter machines [17] and Greibach's blind counter machines [10],

which are also known as integer vector addition systems with states (\mathbb{Z} -VASS) [14]. Furthermore, they can be efficiently translated into each other [1]. The study of Parikh automata on infinite words was initiated just recently by Guha et al. [12], followed by [11]. Closely related to PA on infinite words are blind counter machines on infinite words, as introduced by Fernau and Stiebe [7]. Indeed, these machines turn out to be equivalent to Büchi PA [11]. Another model of automata with counters on infinite words are Büchi VASS. A recent result by Baumann et al. shows that their regular separability problem (that is, the question whether an ω -regular language separates two ω -languages recognized by Büchi VASS) is decidable [2]. Another recent work shows the impact of PA on infinite words. Herrmann et al. [16] introduce a logic subsuming counting MSO and Boolean algebra with Presburger arithmetic, and show decidability over labeled infinite binary trees by introducing Parikh-Muller tree automata (which can be seen as an extension of reachability-regular PA to trees). Yet another recent work by Bergsträßer et al. [3] shows that Ramsey quantifiers in Presburger arithmetic can be eliminated efficiently. Such a quantifier basically asks for the existence of an infinite clique in the graph induced by some formulas. The authors show that their results strengthen recent results on PA on infinite words. We use these results on Ramsey quantifiers to give an explicit application, namely that intersection emptiness of two Büchi PA is decidable, and hence their existential PA model checking problem.

2 PRELIMINARIES

2.1 Finite and infinite words

We write \mathbb{Z} for the set of all integers and \mathbb{N} for the set of non-negative integers including 0. Furthermore, let $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$. Throughout the paper we use the variables c, d, k, ℓ, m, n, z to denote positive integers and we will tacitly assume this without explicit mention. Let Σ be an alphabet, i.e., a finite non-empty set and let Σ^* be the set of all finite words over Σ . For a word $w \in \Sigma^*$, we denote by $|w|$ the length of w , and by $|w|_a$ the number of occurrences of the letter $a \in \Sigma$ in w . We write ε for the empty word of length 0.

An *infinite word* over an alphabet Σ is a function $\alpha : \mathbb{N} \setminus \{0\} \rightarrow \Sigma$. We often write α_i instead of $\alpha(i)$. Thus, we can understand an infinite word as an infinite sequence of symbols $\alpha = \alpha_1\alpha_2\alpha_3\ldots$. For $m \leq n$, we abbreviate the finite infix $\alpha_m \ldots \alpha_n$ by $\alpha[m : n]$. We denote by Σ^ω the set of all infinite words over Σ . We call a subset $L \subseteq \Sigma^\omega$ an ω -*language*. Moreover, for $L \subseteq \Sigma^*$, we define $L^\omega = \{w_1w_2\ldots \mid w_i \in L \setminus \{\varepsilon\}\} \subseteq \Sigma^\omega$. We call a non-empty ω -language $L \subseteq \Sigma^\omega$ *ultimately periodic* if it contains an infinite word of the form uv^ω for finite words $u, v \in \Sigma^*$. We refer to such infinite words also as *ultimately periodic*.

2.2 Regular and ω -regular languages

A *non-deterministic finite automaton* (NFA) is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$, where Q is a finite set of states, Σ is the input alphabet, $q_0 \in Q$ is the initial state, $\Delta \subseteq Q \times \Sigma \times Q$ is the set of transitions and $F \subseteq Q$ is the set of accepting states. We call \mathcal{A} *deterministic* if for every pair $(p, a) \in Q \times \Sigma$ there is exactly one transition of the form $(p, a, q) \in \Delta$ for some $q \in Q$. A *run* of \mathcal{A} on a word $w = w_1 \ldots w_n \in \Sigma^*$ is a (possibly empty) sequence of transitions $r = r_1 \ldots r_n$ with $r_i = (p_{i-1}, w_i, p_i) \in \Delta$ such

that $p_0 = q_0$. We say r is *accepting* if $p_n \in F$. The empty run on ε is accepting if $q_0 \in F$. We define the *language recognized by* \mathcal{A} as $L(\mathcal{A}) = \{w \in \Sigma^* \mid \text{there is an accepting run of } \mathcal{A} \text{ on } w\}$. If a language L is recognized by some NFA \mathcal{A} , we call L *regular*.

A *Büchi automaton* is an NFA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ that takes infinite words as input. A *run* of \mathcal{A} on an infinite word $\alpha_1\alpha_2\alpha_3\dots$ is an infinite sequence of transitions $r = r_1r_2r_3\dots$ with $r_i = (p_{i-1}, \alpha_i, p_i) \in \Delta$ such that $p_0 = q_0$. We say r is *accepting* if there are infinitely many i with $p_i \in F$. We define the ω -*language recognized by* \mathcal{A} as $L_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \text{there is an accepting run of } \mathcal{A} \text{ on } \alpha\}$. If an ω -language L is recognized by some Büchi automaton \mathcal{A} , we call L ω -*regular*. If every state of a Büchi automaton \mathcal{A} is accepting, we call \mathcal{A} a *safety automaton*. Similarly, a *Muller automaton* is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{F})$, where Q, Σ, q_0 , and Δ are defined as for Büchi automata, and $\mathcal{F} \subseteq 2^Q$ is a collection of sets of accepting states. Runs are defined as for Büchi automata, and a run r is accepting if the sets of states that appear infinitely often in r is contained in \mathcal{F} . Deterministic Muller automata have the same expressiveness as non-deterministic Büchi automata [20]. However, deterministic Büchi automata are less expressive than their non-deterministic counterpart [19]. If an ω -language L is recognized by some deterministic Büchi automaton, we call L *deterministic ω -regular*. For a (finite word) language $L \subseteq \Sigma^*$, we define $\vec{L} = \{\alpha \in \Sigma^\omega \mid \alpha[1 : i] \in L \text{ for infinitely many } i\}$. An ω -language L is deterministic ω -regular if and only if $L = \vec{W}$ for a regular language W [19].

2.3 Semi-linear sets

A *linear set* (of dimension $d \geq 1$) is a set of the form $C(b, P) = \{b + p_1z_1 + \dots + p_\ell z_\ell \mid z_1, \dots, z_\ell \in \mathbb{N}\} \subseteq \mathbb{N}^d$, where $b \in \mathbb{N}^d$ and $P = \{p_1, \dots, p_\ell\} \subseteq \mathbb{N}^d$ is a finite set of vectors for some $\ell \geq 0$. We call b the *base vector* and P the set of *period vectors*. A *semi-linear set* is a finite union of linear sets. We also consider semi-linear sets over \mathbb{N}_∞^d , that is, semi-linear sets with an additional symbol ∞ for infinity. As usual, addition of vectors and multiplication of a vector with a number is defined component-wise, where $z + \infty = \infty + z = \infty + \infty = \infty$ for all $z \in \mathbb{N}$, $z \cdot \infty = \infty \cdot z = \infty$ for all $z \geq 1$, and $0 \cdot \infty = \infty \cdot 0 = 0$. For vectors $u = (u_1, \dots, u_c) \in \mathbb{N}_\infty^c$ and $v = (v_1, \dots, v_d) \in \mathbb{N}_\infty^d$, we denote by $u \cdot v = (u_1, \dots, u_c, v_1, \dots, v_d) \in \mathbb{N}_\infty^{c+d}$ the *concatenation of* u and v . We extend this definition to sets of vectors. Let $C \subseteq \mathbb{N}_\infty^c$ and $D \subseteq \mathbb{N}_\infty^d$. Then $C \cdot D = \{u \cdot v \mid u \in C, v \in D\} \subseteq \mathbb{N}_\infty^{c+d}$. We denote by 0^d the d -dimensional all-zero vector, by 1^d the d -dimensional all-one-vector, by e_i^d the d -dimensional vector where the i th entry is 1 and all other entries are 0, and by i_i^d the d -dimensional vector where the i th entry is ∞ and all other entries are 0. We often drop the superscript d if the dimension is clear.

2.4 Parikh-recognizable languages

A *Parikh automaton* (PA) is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ where Q, Σ, q_0 , and F are defined as for NFA, $\Delta \subseteq Q \times \Sigma \times \mathbb{N}^d \times Q$, for some $d \geq 1$, is a finite set of *labeled transitions*, and $C \subseteq \mathbb{N}^d$ is a semi-linear set. We call d the *dimension* of \mathcal{A} and interpret d as a number of *counters*. Analogously to NFA, we call \mathcal{A} *deterministic* if for every pair $(p, a) \in Q \times \Sigma$ there is exactly one labeled transition of the form $(p, a, v, q) \in \Delta$ for some $v \in \mathbb{N}^d$ and $q \in Q$. A *run*

of \mathcal{A} on a word $w = x_1 \dots x_n$ is a (possibly empty) sequence of labeled transitions $r = r_1 \dots r_n$ with $r_i = (p_{i-1}, x_i, v_i, p_i) \in \Delta$ such that $p_0 = q_0$. We define the *extended Parikh image* of a run r as $\rho(r) = \sum_{i \leq n} v_i$ (with the convention that the empty sum equals 0). We say r is accepting if $p_n \in F$ and $\rho(r) \in C$, referring to the latter condition as the *Parikh condition*. The *language recognized by* \mathcal{A} is $L(\mathcal{A}) = \{w \in \Sigma^* \mid \text{there is an accepting run of } \mathcal{A} \text{ on } w\}$.

3 PARIKH AUTOMATA ON INFINITE WORDS

In this section, we recall the acceptance conditions of Parikh automata operating on infinite words that were studied before in the literature.

Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be a PA. A run of \mathcal{A} on an infinite word $\alpha = \alpha_1\alpha_2\alpha_3\dots$ is an infinite sequence of labeled transitions $r = r_1r_2r_3\dots$ with $r_i = (p_{i-1}, \alpha_i, v_i, p_i) \in \Delta$ such that $p_0 = q_0$. The automata defined below differ only in their acceptance conditions, hence they are syntactically equivalent but semantically different (the notion of determinism translates directly). In the following, whenever we say that an automaton \mathcal{A} accepts an infinite word α , we mean that there is an accepting run of \mathcal{A} on α . We begin with the models studied by Guha et al. [12], who also studied the deterministic variants of these models.

- (1) The run r satisfies the *safety condition* if for every $i \geq 0$ we have $p_i \in F$ and $\rho(r_1 \dots r_i) \in C$. We call a PA accepting with the safety condition a *safety PA* [12]. We define the ω -language recognized by a safety PA \mathcal{A} as

$$S_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}.$$

- (2) The run r satisfies the *reachability condition* if there is an $i \geq 1$ such that $p_i \in F$ and $\rho(r_1 \dots r_i) \in C$. We say there is an *accepting hit* in r_i . We call a PA accepting with the reachability condition a *reachability PA* [12]. We define the ω -language recognized by a reachability PA \mathcal{A} as

$$R_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}.$$

- (3) The run r satisfies the *Büchi condition* if there are infinitely many $i \geq 1$ such that $p_i \in F$ and $\rho(r_1 \dots r_i) \in C$. We call a PA accepting with the Büchi condition a *Büchi PA* [12]. We define the ω -language recognized by a Büchi PA \mathcal{A} as

$$B_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}.$$

- (4) The run r satisfies the *co-Büchi condition* if there is i_0 such that for every $i \geq i_0$ we have $p_i \in F$ and $\rho(r_1 \dots r_i) \in C$. We call a PA accepting with the co-Büchi condition a *co-Büchi PA* [12]. We define the ω -language recognized by a co-Büchi PA \mathcal{A} as

$$CB_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}.$$

Furthermore, Guha et al. [12] assume that reachability PA are complete, i.e., for every $(p, a) \in Q \times \Sigma$ there are $v \in \mathbb{N}^d$ and $q \in Q$ such that $(p, a, v, q) \in \Delta$, as incompleteness allows to express additional safety conditions. We also make this assumption in order to study “pure” reachability PA. In fact, we can assume that all models are complete, as the other models can be completed by adding a non-accepting sink. Observe that their deterministic variants are always complete by definition.

Now we recall the models introduced in [11].

- (5) The run satisfies the *reachability and regularity condition* if there is an $i \geq 1$ such that $p_i \in F$ and $\rho(r_1 \dots r_i) \in C$, and there are infinitely many $j \geq 1$ such that $p_j \in F$. We call a PA accepting with the reachability and regularity condition a *reachability-regular PA* [11]. We define the ω -language recognized by a reachability-regular PA \mathcal{A} as

$$RR_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}.$$

- (6) The run satisfies the *limit condition* if there are infinitely many $i \geq 1$ such that $p_i \in F$, and if additionally $\rho(r) \in C$, where the j th component of $\rho(r)$ is computed as follows. If there are infinitely many $i \geq 1$ such that the j th component of v_i has a non-zero value, then the j th component of $\rho(C)$ is ∞ . In other words, if the sum of values in a component diverges, then its value is set to ∞ . Otherwise, the infinite sum yields a positive integer. We call a PA accepting with the limit condition a *limit PA* [11]. We define the ω -language recognized by a limit PA \mathcal{A} as

$$L_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}.$$

- (7) The run satisfies the *strong reset condition* if the following holds. Let $k_0 = 0$ and denote by $k_1 < k_2 < \dots$ the positions of all accepting states in r . Then r is accepting if the sequence k_1, k_2, \dots is infinite and $\rho(r_{k_{i-1}+1} \dots r_{k_i}) \in C$ for all $i \geq 1$. We call a PA accepting with the strong reset condition a *strong reset PA* [11]. We define the ω -language recognized by a strong reset PA \mathcal{A} as

$$SR_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}.$$

- (8) The run satisfies the *weak reset condition* if there are infinitely many reset positions $0 = k_0 < k_1 < k_2 \dots$ such that $p_{k_i} \in F$ and $\rho(r_{k_{i-1}+1} \dots r_{k_i}) \in C$ for all $i \geq 1$. We call a PA accepting with the weak reset condition a *weak reset PA* [11]. We define the ω -language recognized by a weak reset PA \mathcal{A} as

$$WR_\omega(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } \alpha\}.$$

Intuitively worded, whenever a strong reset PA enters an accepting state, the Parikh condition *must* be satisfied. Then the counters are reset. Similarly, a weak reset PA may reset the counters whenever there is an accepting hit, and they must reset infinitely often, too. The equivalence of the (non-deterministic variants of the) two models is shown in [11]. Furthermore, it is shown in [11] that (non-deterministic) limit PA and (non-deterministic) reachability-regular are equivalent, as they recognize exactly ω -languages of the form $\bigcup_i U_i V_i^\omega$, where the U_i are (finite word) languages recognized by PA, and the V_i are regular languages.

4 EXPRESSIVENESS

In this section we study the expressiveness of deterministic PA on infinite words for those models whose deterministic variants were not studied before in the literature. When we show non-inclusions, we always give the strongest separation. By abuse of notation we simply write that a model is a strict/no subset of another model; by

that we mean that the class of ω -languages recognized by the first model is a strict/no subset of the class of ω -languages recognized by the second model.

4.1 ω -regular languages

We begin by showing that every ω -regular language is deterministic limit PA recognizable.

LEMMA 4.1. ω -regular \subseteq deterministic limit PA.

PROOF. Let $L \subseteq \Sigma^\omega$ be ω -regular and let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{F})$ be a deterministic Muller automaton recognizing L . The idea is to construct an equivalent deterministic limit PA $\mathcal{A}' = (Q, \Sigma, q_0, \Delta', Q, C)$ of dimension $|Q|$, where every state is accepting, while encoding the sets in \mathcal{F} into the semi-linear set C . Let $f : Q \rightarrow \{1, \dots, |Q|\}$ be a bijection associating every state with a counter. Hence, we define $\Delta' = \{(p, a, e_{f(q)}^{|Q|}, q) \mid (p, a, q) \in \Delta\}$. For every $F \in \mathcal{F}$, we define $C_F = C(\sum_{q \in F} i_{f(q)}, \{e_{f(q)} \mid q \notin F\})$. That is, for every state in F we expect its counter value to be ∞ , while we expect every other counter value to be a finite number. We choose $C = \bigcup_{F \in \mathcal{F}} C_F$ and hence obtain an equivalent deterministic limit PA.

The strictness is witnessed by the ω -language $\{a^n b^n c^\omega \mid n > 0\}$, which is obviously deterministic limit PA recognizable, but not ω -regular. \square

It turns out that all other models do not recognize all regular ω -languages.

LEMMA 4.2.

$$\omega\text{-regular} \not\subseteq \begin{cases} \text{Deterministic reach-reg. PA} \\ \text{Deterministic Büchi PA} \\ \text{Deterministic strong reset PA} \\ \text{Deterministic weak reset PA} \end{cases}$$

PROOF. None of these PA recognize $L_{a<\infty} = \{\alpha \in \{a,b\}^\omega \mid |\alpha|_a < \infty\}$, which is obviously ω -regular. The proof that these models do not recognize $L_{a<\infty}$ mimics the standard proof showing that this ω -language is not deterministic ω -regular, see e.g. [22]. \square

Observe however these models generalize deterministic Büchi automata. This however is not true for deterministic reach PA, deterministic safety PA nor deterministic co-Büchi PA, as shown in the next lemma.

LEMMA 4.3.

$$\text{Deterministic } \omega\text{-regular} \not\subseteq \begin{cases} \text{Deterministic reach PA} \\ \text{Deterministic safety PA} \\ \text{Deterministic co-Büchi PA} \end{cases}$$

PROOF. As an immediate consequence of Lemma 4.5 (proved below) we have that no deterministic reach PA recognizes the deterministic ω -regular language a^*b^ω .

Guha et al. [12] have shown that (even non-deterministic) safety PA do not recognize the det. ω -regular language $\{a, b\}^\omega \setminus \{a\}^\omega$.

Finally, dually to the previous proof we can show that no co-Büchi PA recognizes the deterministic ω -regular language $L_{a=\infty} = \{\alpha \in \{a, b\}^\omega \mid |\alpha|_a = \infty\}$. \square

4.2 Deterministic Safety PA and co-Büchi PA

As a consequence of Lemma 4.3 we obtain the following corollary.

COROLLARY 4.4.

$$\left. \begin{array}{l} \text{Deterministic reach-reg. PA} \\ \text{Deterministic limit PA} \\ \text{Deterministic Büchi PA} \\ \text{Deterministic strong reset PA} \\ \text{Deterministic weak reset PA} \end{array} \right\} \not\subseteq \left\{ \begin{array}{l} \text{Deterministic safety PA} \\ \text{Deterministic co-Büchi PA} \end{array} \right.$$

As shown in [12] also deterministic reach PA $\not\subseteq$ deterministic safety PA and deterministic reach PA $\not\subseteq$ deterministic co-Büchi PA. Furthermore, the classes of deterministic safety PA and deterministic co-Büchi PA are themselves incomparable as shown in [12].

Vice versa, deterministic safety PA $\not\subseteq$ non-deterministic weak reset PA and deterministic co-Büchi PA $\not\subseteq$ non-deterministic weak reset PA [11]. Hence, these classes are no subclasses of any of the other studied classes.

Overall, deterministic safety PA and deterministic co-Büchi PA are incomparable with all other studied models.

4.3 Deterministic Reach PA

We begin by characterizing the class of deterministic reach PA recognizable ω -languages.

LEMMA 4.5. *An ω -language L is deterministic reach PA recognizable if and only if $L = U\Sigma^\omega$, where $U \subseteq \Sigma^*$ is recognized by a deterministic PA.*

PROOF. Let \mathcal{A} be a deterministic reach PA recognizing L . Then we have $L(\mathcal{A}) = U$. Likewise, if \mathcal{A} is a PA recognizing U , then $R_\omega(\mathcal{A}) = L$ (recall that \mathcal{A} is complete by the definition of determinism). \square

We have the following strict inclusion.

LEMMA 4.6. *Deterministic reach PA \subsetneq deterministic reach-reg. PA.*

PROOF. Let \mathcal{A} be a deterministic reach PA. We may assume that every state of \mathcal{A} is accepting, as we can project the current state into the semi-linear set. To be precise, we introduce two new counters for each state of \mathcal{A} , counting the number of visits and exits. Then, the current state is the (unique) state with one more visit than exit, or in case that the number of visits and exits is the same for every state, then the current state is the initial state of \mathcal{A} . As these statements can be encoded into a semi-linear set, we can assume that every state is equipped with its own semi-linear set, and can hence make every state accepting (and assign the empty set if we want to simulate a non-accepting state). If every state is accepting, then \mathcal{A} is an equivalent deterministic reach-reg. PA.

The strictness is witnessed e.g. by the ω -language $\{a^n b^n a^\omega \mid n > 0\}$, which is deterministic reach-reg. PA recognizable and by Lemma 4.5 not deterministic reach PA recognizable. \square

It remains to show the following incomparability results.

LEMMA 4.7. *Deterministic reach PA $\not\subseteq$ deterministic limit PA.*

PROOF. We show that the deterministic reach PA recognizable ω -language $L = \{\alpha \mid |\alpha[1 : i]|_a = |\alpha[1 : i]|_b \text{ for some } i > 0\}$ is not deterministic limit PA recognizable. The proof is similar to the

proof of Theorem 3 of (the arXiv version of) [12]. Assume there is an n -state deterministic limit PA \mathcal{A} recognizing L . Consider the unique non-accepting run of \mathcal{A} on $a(a^n b^n)^\omega$. Observe that \mathcal{A} visits at least one state twice while reading a b^n -block, and there are at least two of the (infinitely many) b^n -blocks such that \mathcal{A} visits the same state, say q , twice while reading them. Hence, we can shift one such q -cycle to the front and obtain the unique run on an infinite word that is in L . However, this run is still non-accepting, as the extended Parikh image and number of visits of an accepting state do not change. \square

LEMMA 4.8. *Deterministic reach PA $\not\subseteq$ deterministic strong reset PA.*

PROOF. We show that the deterministic reach PA recognizable ω -language $L = \{a^n b^n \mid n \geq 1\} \cdot \{a, b\}^\omega$ is not deterministic strong reset PA recognizable. Assume there is a deterministic strong reset PA \mathcal{A} with n states recognizing L . Let $\alpha = a^n b^\omega$ with unique accepting run $r = r_1 r_2 r_3 \dots$ of \mathcal{A} on α . Let f_1, f_2, \dots be the sequence of reset positions of r and let $i > n$ be minimal such that $i = f_{i'}$ for some $i' \geq 1$ (that is, $f_{i'}$ is the first reset position after reading a b).

First observe that $i < 2n$. Assume that this is not the case. As \mathcal{A} visits at least one state twice while reading b^n , say state q , we observe that \mathcal{A} is caught in a $q \dots q$ cycle while reading b^ω due to determinism. That is, every state that is visited while reading b^ω is already visited while reading the first n many bs . Hence we have $i < 2n$. Now let $j \geq 2n$ be minimal such that $j = f_{j'}$ for some $j' > i'$ is a reset position in r such that the state at position $f_{j'}$ is the same state as the one at position $f_{i'}$ (which exists by the same argument).

Now let $\alpha' = a^n b^{j-n} a^\omega$ with unique accepting run $r' = r'_1 r'_2 r'_3 \dots$ of \mathcal{A} on α' . Observe that $\alpha[1 : j] = \alpha'[1 : j]$, and hence $r[1 : j] = r'[1 : j]$. As the partial runs $r[1 : i]$ and $r[1 : j]$ reach the same accepting state, the run $r[1 : i] r'_{j+1} r'_{j+2} \dots$ is an accepting run of \mathcal{A} on $a^n b^{i-n} a^\omega$. However, as $i - n < n$, this infinite word is not contained in L , a contradiction. \square

4.4 Deterministic Reach-Reg PA

We begin by showing that the every deterministic reach-reg. PA (and hence every deterministic reach PA) can be translated into an equivalent deterministic weak reset PA.

LEMMA 4.9. *Deterministic reach-reg. PA \subseteq deterministic weak reset PA.*

PROOF. Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be a det. reach-reg. PA. Let $\mathcal{A}' = (Q \cup \{q'_0\}, \Sigma, q'_0, \Delta', F, C')$ be a copy of \mathcal{A} with a new fresh initial state q'_0 inheriting all outgoing transitions of q_0 (observe that this modification preserves determinism). We add one new counter that is incremented at every transition leaving q'_0 , and not modified otherwise, that is,

$$\begin{aligned} \Delta' = & \{(p, a, v \cdot 0, q) \mid (p, a, v, q) \in \Delta\} \\ & \cup \{(q'_0, a, v \cdot 1, q) \mid (q_0, a, v, q) \in \Delta\}. \end{aligned}$$

We choose $C' = C \cdot \{1\} \cup \mathbb{N}^d \cdot \{0\}$ and obtain an equivalent weak reset PA \mathcal{A}' .

The strictness is witnessed by the ω -language $\{a^n b^n \mid n > 0\}^\omega$, which is obviously deterministic weak reset PA-recognizable, but not even recognized by (non-deterministic) Büchi PA, which are more expressive than reachability-regular PA [11, 12]. \square

4.5 Deterministic Strong Reset PA

LEMMA 4.10. *Deterministic strong reset PA \subsetneq deterministic weak reset PA.*

PROOF. The inclusion follows by the same argument as in the non-deterministic setting [11] (we use an additional counter to force the weak reset PA to reset whenever an accepting state is visited).

The strictness follows from the fact that $\{a^n b^n \mid n \geq 1\} \cdot \{a, b\}^\omega$ is deterministic reach PA recognizable, and hence deterministic weak reset PA recognizable (by Lemma 4.6 and Lemma 4.9), but not recognized by any deterministic strong reset PA, as shown in Lemma 4.8. \square

LEMMA 4.11. *Deterministic strong reset PA $\not\subseteq$ deterministic Büchi PA.*

PROOF. The argument is as in Lemma 4.9. The ω -language $\{a^n b^n \mid n > 0\}^\omega$ is deterministic strong reset PA recognizable, but there is no Büchi PA recognizing it [12]. \square

LEMMA 4.12. *Deterministic strong reset PA $\not\subseteq$ deterministic limit PA.*

PROOF. This follows from the previous proof as limit PA are less expressive than Büchi PA [11]. \square

4.6 Deterministic Büchi PA

We show that ω -languages recognized by deterministic Büchi PA can be characterized in a similar way as deterministic ω -regular languages.

LEMMA 4.13. *An ω -language L is deterministic Büchi PA recognizable if and only if $L = \vec{P}$ where P is recognized by a deterministic PA.*

PROOF. Let \mathcal{A} be a deterministic Büchi PA recognizing L and let $\alpha \in B_\omega(\mathcal{A})$ with accepting run r . As r has infinitely many accepting hits by definition, we have $\alpha \in L(\vec{\mathcal{A}})$. Similarly, let \mathcal{A} be a deterministic PA recognizing P and let $\alpha \in \vec{P}$. As \mathcal{A} is deterministic, the unique run of \mathcal{A} on α has infinitely many accepting hits, hence we have $\alpha \in B_\omega(\mathcal{A})$. \square

LEMMA 4.14. *Deterministic Büchi PA $\not\subseteq$ deterministic limit PA.*

PROOF. The proof is almost identical to the proof of Lemma 4.7, but this time we consider the ω -language $L_{a=b} = \{\alpha \mid |\alpha[1 : i]|_a = |\alpha[1 : i]|_b \text{ for } \infty \text{ many } i\}$. Then we can re-use the same argument as the constructed infinite word has indeed infinitely many balanced a - b prefixes. \square

LEMMA 4.15. *Deterministic Büchi PA $\not\subseteq$ deterministic weak reset PA.*

PROOF. Let $L_{a=b}$ be as in the last proof and similarly define $L_{a=c} = \{\alpha \mid |\alpha[1 : i]|_a = |\alpha[1 : i]|_c \text{ for } \infty \text{ many } i\}$. We show that the deterministic Büchi PA recognizable ω -language $L_{a=b} \cup L_{a=c}$ is not deterministic weak reset PA recognizable. Assume there is a deterministic weak reset PA \mathcal{A} recognizing $L_{a=b} \cup L_{a=c}$. Consider

the unique accepting run r of \mathcal{A} on $\alpha = (ab)^\omega$ and let i, j be two positions with $i + 1 < j$ such that r resets after reading $\alpha[1 : i]$ and $\alpha[1 : j]$ in the same state (such a pair of positions does always exist by the infinite pigeonhole principle). Now consider the infinite word $\alpha[1 : i]c^{|\alpha[1 : i]|_a}(ac)^\omega$, which is also accepted by \mathcal{A} . However, this implies that \mathcal{A} also accepts $\alpha[1 : j]c^{|\alpha[1 : i]|_a}(ac)^\omega$, as \mathcal{A} is in the same (accepting) state after reading $\alpha[1 : i]$ as well as $\alpha[1 : j]$, but this infinite word is not contained in $L_{a=b} \cup L_{a=c}$, as $\alpha[1 : j]$ contains at least one more a than $\alpha[1 : i]$, a contradiction. \square

We note however that the class of ω -languages recognized by deterministic Büchi PA with a linear set form a subclass of the class of ω -languages recognized by deterministic weak reset PA with a linear set, as clarified in the following lemma.

LEMMA 4.16. *Let \mathcal{A} be a deterministic Büchi PA with a linear set $C(b, P)$. Then there is an equivalent deterministic weak reset PA.*

PROOF. First we observe that if $b = 0$, then we have $B_\omega(\mathcal{A}) = WR_\omega(\mathcal{A})$. To see this, let $\alpha \in B_\omega(\mathcal{A})$ with (unique) accepting run r . By Dickson's Lemma [6], the run r contains an infinite monotone sequence $s_1 < s_2 < \dots$ of accepting hits, that is, for all $i \geq 0$ we have $\rho(r[1 : s_i]) \in C(b, P)$ and for all $j > i$ we have $\rho(r[s_i + 1 : s_j]) \in C(b, P)$. Hence, the run r also satisfies the weak reset condition. For the other direction let $\alpha \in WR_\omega(\mathcal{A})$ with (unique) accepting run r and reset positions $0 = k_0 < k_1 < k_2 \dots$. As we assume $b = 0$, it is immediate that $\rho(r[1 : k_i]) \in C(b, P)$ for all $i \geq 1$. Hence, the run r also satisfies the Büchi condition.

Finally we argue that we can always assume that $b = 0$. Indeed, we can always encode b into the state space of \mathcal{A}_1 . \square

5 CLOSURE PROPERTIES

We study the closure properties of the deterministic variants of the models introduced by Grobler et al. [11], that is, deterministic limit PA, deterministic reachability-regular PA, deterministic strong reset PA, and deterministic weak reset PA.

It is well known that semi-linear sets over \mathbb{N}^d are closed under complement [13]. Before we study deterministic limit automata we show that this is also true for semi-linear sets enriched with ∞ .

LEMMA 5.1. *Let $C \subseteq \mathbb{N}_\infty^d$ be a semi-linear set. Then the complement $\bar{C} = \mathbb{N}_\infty^d \setminus C$ is semi-linear.*

PROOF. Let $f : \mathbb{N}_\infty \rightarrow \mathbb{N}$ be the bijection with $f(\infty) = 0$ and $f(i) = i + 1$ for $i \in \mathbb{N}$. We extend f to vectors $v = (v_1, \dots, v_d) \in \mathbb{N}_\infty^d$ and sets of vectors $C \subseteq \mathbb{N}_\infty^d$ component-wise: $f(v_1, \dots, v_d) = (f(v_1), \dots, f(v_d))$ and $f(C) = \{f(v) \mid v \in C\}$. Note that $f(C) \subseteq \mathbb{N}^d$.

Now we observe that $f(C)$ is semi-linear if and only if C is semi-linear. First assume that C is semi-linear. We may assume that $C = C(b, P)$ is linear, as we can carry out the following procedure for every linear set individually. Assume $b = (b_1, \dots, b_d)$. We define $D_\infty(b) = \{i \mid b_i = \infty\}$. For a set $D \subseteq \{1, \dots, d\}$ with $D_\infty(b) \subseteq D$ and a vector $v = (v_1, \dots, v_d) \in \mathbb{N}_\infty^d$, let $v^D = (v_1^D, \dots, v_d^D)$ with $v_i^D = 0$ if $i \in D$ and $v_i^D = v_i$ if $i \notin D$. Furthermore, we call a subset $P' \subseteq P$ of period vectors D -compatible if for every $i \in D \setminus D_\infty(b)$, the set P' contains at least one vector where the i th component is ∞ , and if for every $i \notin D$, the set P' contains no vector where the i th component is ∞ . Observe that this definition ensures that for

every vector $v \in P'$ we have $v^D \in \mathbb{N}^d$, that is, no component in v^D is ∞ . Let $\mathcal{P}_\infty^D \subseteq 2^P$ be the collection of D -compatible subsets of P . Then we have $f(C) = \bigcup_{D_\infty(b) \subseteq D \subseteq \{1, \dots, d\}} \bigcup_{P' \in \mathcal{P}_\infty^D} C(b^D + 1^D + \sum_{p \in P'} p^D, \{p^D \mid p \in P'\})$, which is semi-linear by definition.

For the other direction, we may again assume that $f(C) = C(b, P)$ is linear. Similar to above, assume $b = (b_1, \dots, b_d)$ and define $D_0(b) = \{i \mid b_i = 0\}$. For a set $D \subseteq D_0(b)$ we call a subset $P' \subseteq P$ of period vectors D -safe if for every $i \in D$ the set P' contains no vector where the i th component is not 0, and if for every $i \notin D$ the set P' contains at least one vector where i th component is not 0. Let $\mathcal{P}_0^D \subseteq 2^P$ be the collection of D -safe subsets of P . Let $i_D = (v_1, \dots, v_d)$ with $v_i = \infty$ if $i \in D$ and $v_i = 0$ if $i \notin D$. Then we have $C = \bigcup_{D \subseteq D_0(b)} \bigcup_{P' \in \mathcal{P}_0^D} C(i_D + b - 1 + \sum_{p \in P'} p, P')$, which is semi-linear by definition (observe that every component in $i_D + b + \sum_{p \in P'} p$ is strictly greater 0, hence we can subtract 1 without getting negative; furthermore, we assume $\infty - 1 = \infty$).

As semi-linear sets over \mathbb{N}^d are closed under complement, we have C is semi-linear iff $f(C)$ is semi-linear iff $\mathbb{N}^d \setminus f(C)$ is semi-linear iff $f^{-1}(\mathbb{N}^d \setminus f(C)) = \bar{C}$ is semi-linear. \square

LEMMA 5.2. *The class of deterministic limit PA recognizable languages is closed under union, intersection and complement.*

PROOF. First observe that we can always assume that every state of a (deterministic) limit PA is accepting, as we can check the existence of an accepting state being visited infinitely often in the semi-linear set. To achieve this, we introduce one new counter and increment it at every transition that points to an accepting state. In the semi-linear set we enforce that this counter is ∞ .

Hence, we can show the closure under union and intersection by a standard product construction. In case of union, we test whether at least one automaton has good counter values, and we can show the closure under intersection by testing whether both automata have good counter values. Closure under complement follows immediately from Lemma 5.1. \square

LEMMA 5.3. *The class of deterministic reach-reg. PA recognizable languages is not closed under union, intersection or complement.*

PROOF. First we show non-closure under union. Let

$$L_1 = \{uca \mid u \in \{a, b, c\}^*, |u|_a = |u|_b, |\alpha|_c = \infty\}$$

and

$$L_2 = \{vab \mid v \in \{a, b, c\}^*, |v|_b = |v|_c, |\beta|_a = \infty\}.$$

Both languages are deterministic reach-reg. PA recognizable, as witnessed by the automaton in Figure 1 and the fact that L_2 can be obtained from L_1 by shuffling symbols. We show that the language $L_1 \cup L_2$ is not deterministic reach-reg. PA recognizable. Assume there is an n -state deterministic reach-reg. PA \mathcal{A} recognizing L . Then there is a unique accepting run $r_1 r_2 \dots$ of \mathcal{A} on abc^ω . In particular, at some point the automaton verifies the Parikh condition, say after using the transition r_i . Let $m = \max\{n + 1, i\}$ and consider the infinite word $abc^m b^{m-1} a^\omega$ with the unique accepting run $r'_1 r'_2 \dots$ of \mathcal{A} . Due to determinism, we have $r_j = r'_j$ for all $j \leq m + 2$; in particular, the automaton verifies the Parikh condition within this run prefix. As $m - 1 \geq n$, the automaton visits a state, say q

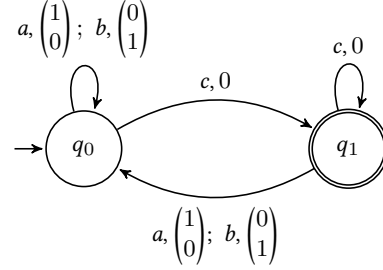


Figure 1: The deterministic reach-reg. PA with $C = C(0, \{1\})$ for $L_1 = \{uca \mid u \in \{a, b, c\}^*, |u|_a = |u|_b, |\alpha|_c = \infty\}$.

twice while reading b^{m-1} . Hence, we can pump this q -cycle and obtain an accepting run of \mathcal{A} on $abc^m b^{m-1+k} a^\omega$ for some $k > 0$, a contradiction.

To show the non-closure under intersection, define $L_1 = \{\alpha \mid |\alpha[1 : i]|_a = |\alpha[1 : i]|_b \text{ for some } i\}$ and $L_2 = \{\alpha \mid |\alpha[1 : i]|_a = |\alpha[1 : i]|_c \text{ for some } i\}$. Suppose there is an n -state deterministic reach-reg. PA recognizing $L_1 \cap L_2$. Let $\alpha = a(a^n b^n)^{n+1} c^{n(n+1)+1} a^\omega$. The unique run r of \mathcal{A} on α is not accepting, as α has no balanced a - b prefix. Observe that \mathcal{A} visits at least one state twice while reading a b^n -block. Furthermore, there is a state, say q , such that \mathcal{A} visits q twice while reading two different b^n -blocks, as there are $n + 1$ different b^n -blocks. Hence, we can swap the latter q -cycle to the front and obtain an infinite word, say α' in $L_1 \cap L_2$, with a unique accepting run r' of \mathcal{A} . This run verifies the Parikh condition at some point. We distinguish two cases. If r' verifies the Parikh condition before reading the first c , we can depump the $c^{n(n+1)+1}$ -block and obtain an accepting run on an infinite word without a balanced a - c -prefix, a contradiction. Hence assume that r' verifies the Parikh condition after reading at least one c , say at position k . However, then we have $\rho(r[1 : k]) = \rho(r'[1 : k])$, and hence \mathcal{A} also accepts α , a contradiction.

The non-closure under complement is witnessed by $L_{a < \infty}$, see Lemma 4.2. \square

LEMMA 5.4. *The class of deterministic weak reset PA recognizable languages is not closed under union, intersection or complement.*

PROOF. We begin with the non-closure under union. The ω -languages $L_{a=b}$ and $L_{a=c}$ (as defined in Lemma 4.15) are both deterministic weak reset PA recognizable. As shown in Lemma 4.15, their union is not.

The argument for the non-closure under intersection is the same as for the non-deterministic setting [11], which follows from [7] and [12]: Let $L_{2a=b} = \{\alpha \mid 2|\alpha[1 : i]|_a = |\alpha[1 : i]|_b \text{ for } \infty \text{ many } i\}$. Then $L_{a=b} \cap L_{2a=b}$ is not even ultimately periodic, and hence not recognized by any weak reset PA, as they only recognize ultimately periodic ω -languages [11].

The non-closure under complement is again witnessed by $L_{b < \infty}$, see Lemma 4.2. \square

LEMMA 5.5. *The class of deterministic strong reset PA recognizable languages is not closed under union, intersection or complement.*

PROOF. We begin with the non-closure under union. Let $L = \{c^n a^n c^n b^n \mid n > 0\}^\omega$ and $L_{c=\infty} = \{\alpha \mid |\alpha|_c = \infty\}$. Observe that $a^n c^\omega \in L \cup L_{c=\infty}$ for every $n \geq 0$. Assume there is a deterministic strong reset PA recognizing $L \cup L_{c=\infty}$. Let $n_1 \neq n_2$ be such that the unique accepting runs of \mathcal{A} on $\alpha_1 = a^{n_1} c^\omega$ resp. $\alpha_2 = a^{n_2} c^\omega$ reset in the same state the first time they reset after reading at least one c , say after reading $\alpha_1[1 : i_1]$ resp. $\alpha_2[1 : i_2]$ (with $i_1 > n_1$ and $i_2 > n_2$). As $a^{n_1} c^{n_1-i_1} b^{n_1} (ab)^\omega$ is also accepted by \mathcal{A} , the infinite word $a^{n_2} c^{n_2-i_2} b^{n_1} (ab)^\omega$ is also accepted by \mathcal{A} , a contradiction.

To show the non-closure under intersection, we use an argument similar to the non-deterministic setting. Let $L_1 = \{a^n b^n \mid n > 0\}^\omega$ and $L_2 = \{a^n b^{2n} \mid n > 0\}$. Then $L_1 \cap L_2$ contains only one infinite word, namely $aba^2b^2a^4b^4 \dots$. Hence $L_1 \cap L_2$ is not ultimately periodic and hence not accepted by any strong reset PA [11].

The non-closure under complement again follows from Lemma 4.2. \square

Finally we remark that deterministic reach-reg. PA, deterministic limit PA, deterministic strong reset PA and deterministic weak reset PA are strictly weaker than their non-deterministic counterparts. This follows immediately from their different closure properties: reach-reg. PA, weak reset PA (and hence strong reset PA) are closed under union, and limit PA are not closed under complement [11]. The authors do not explicitly mention that limit PA are not closed under complement. This however follows from their characterization of limit PA and the fact (non-deterministic finite word) PA are not closed under complement [18].

6 DECISION PROBLEMS AND MODEL CHECKING

6.1 Decision Problems

In this section, we consider the following decision problems for PA on infinite words.

- **Emptiness:** given PA \mathcal{A} , is the ω -language of \mathcal{A} empty?
- **Membership:** given PA \mathcal{A} and finite words u, v , does \mathcal{A} accept uv^ω ?
- **Universality:** given PA \mathcal{A} , does \mathcal{A} accept every infinite word?

LEMMA 6.1. *Emptiness for deterministic limit PA, deterministic reach-reg. PA, deterministic weak reset PA and deterministic strong reset PA is coNP-complete.*

PROOF. Containment in coNP for all models is witnessed by the fact that testing non-emptiness for non-deterministic reset PA is in NP, and that they generalize all of these models (this follows immediately from [11]).

Hardness is very similar to the finite word case [8], as we can reduce subset sum to non-emptiness for all of these models. \square

LEMMA 6.2. *Membership for deterministic limit PA, deterministic reach-reg. PA, deterministic weak reset PA and deterministic strong reset PA is NP-complete.*

PROOF. As membership for the non-deterministic counterparts of these models is in NP [11], the upper bound follows immediately.

Hardness follows from a simple reduction from the membership problem for semi-linear sets, that is the question, given a semi-linear set $C \subseteq \mathbb{N}^d$ and a vector $v \in \mathbb{N}^d$, is $v \in C$? This problem is known to be NP-complete [13]. \square

LEMMA 6.3. *Universality for deterministic limit PA and deterministic strong reset PA is decidable, while universality for deterministic reach-reg. PA and deterministic weak reset PA is undecidable.*

PROOF. The decidability result for deterministic limit PA follows from the fact that a (deterministic limit) PA \mathcal{A} is universal if and only if $\overline{L_\omega(\mathcal{A})}$ is empty. Now, deterministic limit PA are closed under complement by Lemma 5.2 and emptiness is decidable by Lemma 6.1.

In contrast Guha et al. have shown that universality is already undecidable for deterministic reach PA [12]. Hence, the undecidability for deterministic reach-reg. PA and deterministic weak reset PA follows from Lemmas 4.6 and 4.9.

Finally we show that universality for deterministic strong reset PA is decidable by reducing it to checking emptiness of the complement ω -language. Let \mathcal{A} be a deterministic strong reset PA. Observe that \mathcal{A} rejects an infinite word α whenever one of the following two conditions is met:

- (1) The unique run of \mathcal{A} on α visits every accepting state just finitely often.
- (2) The unique run of \mathcal{A} on α visits once an accepting state with bad counter values.

The first condition is ω -regular, while the second one can be tested using reach PA. As the union of an ω -regular language and a reach PA recognizable language is recognized by a reach-reg. PA, for which emptiness is decidable [11], we obtain the desired result.

Let us begin by showing how a reach PA can test whether a deterministic strong reset PA \mathcal{A} rejects at least one infinite word α because the unique run r of \mathcal{A} on α reaches an accepting state with bad counter values. Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ and assume that every state in \mathcal{A} is reachable from the initial state (otherwise they can safely be removed), and that $q_0 \notin F$ (this can be done by adding a fresh initial state while preserving determinism). Say $r = r_0 r_1 r_2 \dots$ and $0 = k_0 < k_1 < k_2 \dots$ are the reset positions. Given that $k_0, k_1, k_2 \dots$ is an infinite sequence, there is $j \geq 0$ such that the partial run $r_{k_j+1} \dots r_{k_{j+1}}$ collects a vector not in C . We re-use the following idea from [11, Lemma 30]. For any two states $p, q \in Q$ let $\mathcal{A}_{p \Rightarrow q} = (Q \cup \{q'_0\}, \Sigma, q'_0, \Delta_{p \Rightarrow q}, \{q\}, C)$ where $\Delta_{p \Rightarrow q} = \{(q_1, a, v, q_2) \mid (q_1, a, v, q_2) \in \Delta, q_1 \notin F\} \cup \{(q'_0, a, v, q_2) \mid (p, a, v, q_2) \in \Delta\}$. Hence, if we interpret $\mathcal{A}_{p \Rightarrow q}$ as a finite word PA, then it accepts words all accepted by the finite word PA \mathcal{A} when starting in p , ending in q , and not visiting an accepting state in-between. In other words, if $p, q \in F$, then $\mathcal{A}_{p \Rightarrow q}$ accepts all finite infixes that the strong reset PA \mathcal{A} may read when the last reset was in p , and the next reset is in q . Now let $\bar{\mathcal{A}}_{p \Rightarrow q}$ be defined as $\mathcal{A}_{p \Rightarrow q}$, but C is complemented. Consider for all $p, q \in F$ the finite-word PA $\bar{\mathcal{A}}_{p \Rightarrow q}$. If at least one of these PA accepts at least one word, then the strong reset PA \mathcal{A} is not universal. To see this, assume that $\bar{\mathcal{A}}_{p \Rightarrow q}$ accepts a word $w \in \Sigma^*$. Furthermore, let $v \in \Sigma^*$ be an arbitrary word such that (the unique run of) \mathcal{A} is in state p after reading v . As p is an accepting state which enforces a reset, the counters of \mathcal{A} are all 0 after reading v . Now, if we feed w to \mathcal{A} , the

automaton reaches the accepting state q (without seeing any other accepting state in between by the definition of $\bar{\mathcal{A}}_{p \Rightarrow q}$), and hence rejects at this point, as the counter values are bad. Hence, \mathcal{A} rejects any infinite word with prefix vw , and is hence not universal.

Hence, for every $p, q \in F$ we construct a (non-deterministic) reach PA that accepts all infinite words of the form $vw\Sigma^\omega$ such that \mathcal{A} is in state p after reading v (there might be several resets before reaching p), and $w \in L(\bar{\mathcal{A}}_{p \Rightarrow q})$. Finally, we build the union of all these automata for all $p, q \in F$ and obtain a reach PA \mathcal{A}_\cup whose ω -language is not empty if and only if there is an infinite word such that the unique run of \mathcal{A} visits an accepting state with bad counter values (Condition 2 of the list above).

Let \mathcal{B} be the Büchi automaton obtained from \mathcal{A} by forgetting all vectors and let $\bar{\mathcal{B}}$ be a Büchi automaton recognizing the complement of $L_\omega(\mathcal{B})$.

Then \mathcal{A} is universal if and only if $R_\omega(\mathcal{A}_\cup) \cup L_\omega(\bar{\mathcal{B}})$ is empty, which can be tested by turning both automata into reach-reg. PA, which are trivially closed under union [11, Theorem 9]. \square

6.2 Model Checking

In the classical model checking problem, we are given a system \mathcal{K} and a specification, e.g. represented as an automaton \mathcal{A} , and the question is whether every run of \mathcal{K} satisfies the specification, i.e., we ask $L(\mathcal{K}) \subseteq L(\mathcal{A})$, which can be equivalently written as $L(\mathcal{K}) \cap \overline{L(\mathcal{A})} = \emptyset$. However, as complementing is often expensive or not even possible, another approach is to specify the set of all bad runs and ask whether no run of \mathcal{K} is bad, which boils down to the question is $L(\mathcal{K}) \cap L(\mathcal{A}) = \emptyset$? We call the first approach *universal model checking* and the latter approach *existential model checking*. In our setting we assume the specification \mathcal{A} to be a PA operating on infinite words, while the system \mathcal{K} may be given as a Kripke-structure (which can be seen as a safety automaton [4]), in which case the goal is to solve *safety model checking*, or also as a PA operating on infinite words, in which case the goal is to solve *PA model checking*. Hence, we consider four problems in total.

As shown in [11], the existential safety model checking problem for almost all (non-deterministic variants of the) models considered in this paper is decidable in coNP, the exception being safety PA and co-Büchi PA.

THEOREM 6.4 ([11]). *Existential safety model checking is coNP-complete for reach PA, reach-reg. PA, limit PA Büchi PA, strong reset PA, and weak reset PA.*

Hence, this problem is also decidable in coNP for their deterministic variants. The matching lower-bounds follow from an easy reduction from their emptiness problems by asking whether $\Sigma^\omega \cap L(\mathcal{A}) = \emptyset$ (a safety automaton for Σ^ω is easily found).

COROLLARY 6.5. *Existential safety model checking is coNP-complete for deterministic reach PA, deterministic reach-reg. PA, deterministic limit PA Büchi PA, deterministic strong reset PA, and deterministic weak reset PA.*

Guha et al. [12] have shown that emptiness is undecidable for deterministic safety PA and deterministic co-Büchi PA. Hence, their existential safety model checking problem is undecidable.

COROLLARY 6.6. *Existential safety model checking is undecidable for deterministic safety PA and deterministic co-Büchi PA.*

Similarly, the universal model checking problems are undecidable for all models that already have an undecidable universal-ity problem, as we can easily reduce from universality by asking whether $\Sigma^\omega \subseteq L(\mathcal{A})$. Hence, we obtain the following corollary as an immediate consequence of Lemma 6.3 and the results in [12].

COROLLARY 6.7. *Universal safety model checking and universal PA model checking are undecidable for deterministic reach PA, deterministic reach-reg. PA, deterministic Büchi PA and deterministic weak reset PA.*

Guha et al. [12] have shown that the universal PA model checking problem is decidable for deterministic safety PA and deterministic co-Büchi PA. We note that this is also the case for deterministic limit PA, which follows from their closure properties. Furthermore, we show that the universal PA model checking problem is decidable for deterministic strong reset PA. As shown in Lemma 6.3, the complement ω -languages of deterministic strong reset PA are recognized by reach-reg. PA. Hence, given two deterministic strong reset PA $\mathcal{A}_1, \mathcal{A}_2$, the question whether $SR_\omega(\mathcal{A}_1) \subseteq SR_\omega(\mathcal{A}_2)$ (which is equivalent to the question whether $SR_\omega(\mathcal{A}_1) \cap \overline{SR_\omega(\mathcal{A}_2)} = \emptyset$) boils down to deciding intersection emptiness of a deterministic strong reset PA and a reach-reg. PA.

We first show that deciding intersection emptiness of a deterministic strong reset PA recognizable ω -language and an ω -regular language is decidable. To achieve that, we begin by showing that the intersection of a deterministic strong reset PA recognizable ω -language and an ω -regular language is still ultimately periodic.

LEMMA 6.8. *Let $\mathcal{A} = (Q_1, \Sigma, p_I, \Delta_1, F_1, C)$ be a deterministic strong reset PA and $\mathcal{B} = (Q_2, \Sigma, q_I, \Delta_2, F_2)$ be a Büchi automaton. Then $SR_\omega(\mathcal{A}) \cap L_\omega(\mathcal{B})$ is ultimately periodic.*

PROOF. Assume $SR_\omega(\mathcal{A}) \cap L_\omega(\mathcal{B}) \neq \emptyset$ and let α be an infinite word accepted by both automata, \mathcal{A} and \mathcal{B} . If $\alpha = uv^\omega$ for some $u, v \in \Sigma^*$, we are done. Hence assume that this is not the case. Let $\mathcal{A} = (Q_1 \times Q_2, \Sigma, (p_I, q_I), \Delta, F_1 \times F_2, C)$ with $\Delta = \{(p, q), a, v, (p', q') \mid (p, a, v, p') \in \Delta_1, (q, a, q') \in \Delta_2\}$ be the product automaton¹ of \mathcal{A} and \mathcal{B} . As $\alpha \in SR_\omega(\mathcal{A})$, the unique accepting run $(p_0, \alpha_1, v_1, p_1) (p_1, \alpha_2, v_2, p_2) \dots$ with $p_0 = p_I$ is accepting. Likewise, as $\alpha \in L_\omega(\mathcal{B})$, there is an accepting run $(q_0, \alpha_1, q_1) (q_1, \alpha_2, q_2) \dots$ with $q_0 = q_I$ of \mathcal{B} on α . Hence, $r = ((p_0, q_0), \alpha_1, v_1, (p_1, q_1)) ((p_1, q_1), \alpha_2, v_2, (p_2, q_2)) \dots$ is a run of \mathcal{A} on α with the following properties:

- there is $p_f \in F_1$ such that for infinitely many i we have $p_i = p_f$. Let f_1, f_2, \dots denote the positions of all occurrences of a state of the form (p_f, \cdot) in r .
- there is $q_f \in F_2$ such that for infinitely many i we have $q_i = q_f$.

Let $j \geq f_1$ be minimal such that $q_j = q_f$. Now let $k \leq j$ be maximal such that $k = f_\ell$ for some $\ell \geq 1$. Then $r[1 : f_\ell]r[f_\ell + 1 : f_{\ell+1}]^\omega$ is an accepting run of \mathcal{A} on an ultimately periodic word, say uv^ω , that visits an accepting state of \mathcal{B} infinitely often. Hence $uv^\omega \in SR_\omega(\mathcal{A}) \cap L_\omega(\mathcal{B})$. \square

¹We note that \mathcal{A} interpreted as a strong reset PA does not recognize $SR_\omega(\mathcal{A}) \cap L_\omega(\mathcal{B})$. Instead, it accepts all infinite words $\alpha \in SR_\omega(\mathcal{A})$ such that \mathcal{B} has an infinite but not necessarily accepting run on α .

LEMMA 6.9. *Let \mathcal{A} be a deterministic strong reset PA and let \mathcal{B} be a Büchi automaton. The question $SR_\omega(\mathcal{A}) \cap L_\omega(\mathcal{B}) = \emptyset$ is coNP-complete.*

PROOF. Hardness is obvious; hence, we focus on the containment in coNP. By the previous lemma it is sufficient to check the existence of an ultimately periodic word. Let $\mathcal{A} = (Q_1, \Sigma, p_0, \Delta_1, F_1, C)$ and $\mathcal{B} = (Q_2, \Sigma, q_0, \Delta_2, F_2)$. Recall the algorithm in [11, Lemma 30] that decides the non-emptiness problem for reset PA in NP by exploiting that every non-empty ω -language accepted by a strong reset PA contains an ultimately periodic word. This algorithm basically guesses an accepting state p of \mathcal{A} that is seen infinitely often, and a sequence of distinct accepting states $p_1 \dots p_n$ such that $p_i = p$ for some $i \leq n$. Then, the algorithm tests for all $0 \leq i < n$ whether the finite word PA $\mathcal{A}_{p_i \Rightarrow p_{i+1}}$ accepts at least one word, and whether $\mathcal{A}_{p_n \Rightarrow p}$ accept at least one word (where $\mathcal{A}_{p \Rightarrow q}$ is defined as in the proof of Lemma 6.3). We modify the algorithm as follows.

First, let \mathcal{A} be the product automaton of \mathcal{A} and \mathcal{B} (as in the previous proof; however, the set of accepting states is not important for the algorithm). We guess a state $q \in F_2$ that we expect to be seen infinitely often to satisfy the acceptance condition of \mathcal{B} . Furthermore, similar to the algorithm above we guess a sequence of distinct states $(p_1, q_1)(p_2, q_2) \dots (p_n, q_n)$ such that for some $\ell \leq n$ we have $q_\ell = q$, for some $i \leq \ell$ we have $p_i = p$, and for all $j \neq \ell$ we have that p_j is accepting in \mathcal{A} . If the sequence contains a state $(p_i, q_i) \in F_1 \times F_2$, we can interpret \mathcal{A} as a reset PA with accepting states $F_1 \times Q_2$ and run the algorithm, as the Büchi condition is automatically satisfied.

Hence assume that $p_\ell \notin F_1$ and $q_j \notin F_2$ for all $j \neq \ell$. Let $\mathcal{A}_{p_{\ell-1} \xrightarrow{q_\ell} p_{\ell+1}}$ be the finite word PA that accepts all finite infixes accepted by the product automaton \mathcal{A} when starting in $(p_{\ell-1}, q_{\ell-1})$, visiting (p_ℓ, q_ℓ) at some point, and ending in $(p_{\ell+1}, q_{\ell+1})$ such that for all internal states (p_i, q_i) we have $p_i \notin F_1$. Two achieve this, we take two copies of \mathcal{A} , where all accepting states in the first copy are not reachable, all accepting states in the second copy have no outgoing transitions, and the second copy can only be reached from the first copy via (p_ℓ, q_ℓ) .

Formally, let

$$\mathcal{A}_{p_{\ell-1} \xrightarrow{q_\ell} p_{\ell+1}} = (Q_1 \times Q_2 \times \{1, 2\} \cup \{q'_0\}, \Sigma, q'_0, \Delta', \{(p_{\ell+1}, q_{\ell+1})\}, C)$$

with

$$\begin{aligned} \Delta' = & \{((p, q, 1), a, v, (p', q', 1)) \mid ((p, q), a, v, (p', q')) \in \Delta, p, p' \notin F_1\} \\ & \cup \{((p, q, 2), a, v, (p', q', 2)) \mid ((p, q), a, v, (p', q')) \in \Delta, p \notin F_1\} \\ & \cup \{((p, q, 1), a, v, (p_\ell, q_\ell, 2)) \mid ((p, q), a, v, (p_\ell, q_\ell)) \in \Delta\} \\ & \cup \{(q'_0, a, v, (p', q', 1)) \mid ((p_{\ell-1}, q_{\ell-1}), a, v, (p', q')) \in \Delta, p' \notin F_1\} \\ & \cup \{(q'_0, a, v, (p_\ell, q_\ell, 2)) \mid ((p_{\ell-1}, q_{\ell-1}), a, v, (p_\ell, q_\ell)) \in \Delta\}. \end{aligned}$$

Now the algorithm is similar to the original non-emptiness algorithm for reset PA. For every $0 \leq j < \ell - 1$ and $\ell + 1 \leq j < n$ we test $\mathcal{A}_{(p_j, q_j) \Rightarrow (p_{j+1}, q_{j+1})}$ as well as $\mathcal{A}_{(p_n, q_n) \Rightarrow (p_i, q_i)}$ for non-emptiness. Furthermore, we test $\mathcal{A}_{p_{\ell-1} \xrightarrow{q_\ell} p_{\ell+1}}$ for non-emptiness.

If all these automata accept at least one word, we conclude that $SR_\omega(\mathcal{A}) \cap L_\omega(\mathcal{B}) \neq \emptyset$, as the guess of our sequence implies that at least one accepting state of \mathcal{B} is seen infinitely often, while the rest follows from the correctness of the algorithm in [11]. \square

In order to show that testing intersection emptiness of a deterministic strong reset PA and a reach-reg. PA is decidable, we combine the algorithm in the previous proof with the NP-algorithm in [15] deciding the reachability problem for \mathbb{Z} -VASS with k nested zero-tests (\mathbb{Z} -VASS $_k^{\text{nz}}$). A \mathbb{Z} -VASS $_k^{\text{nz}}$ (of dimension $d \geq 1$) is a tuple $V = (Q, Z, E)$ where Q is a finite set of states, $Z \subseteq \{0, 1, \dots, d\}$ is its set of zero tests with $|Z \setminus \{0\}| = k$, and $E \subseteq Q \times \mathbb{Z}^d \times Z \times Q$ is a finite set of transitions. A configuration of V is a pair $(p, v) \in Q \times \mathbb{Z}^d$. Assume $v = (v_1, \dots, v_d)$. We write $(p, v) \vdash_V (p', v')$ if there is a transition $(p, u, \ell, p') \in E$ such that $v' = v + u$ and $v_1 = \dots = v_\ell = 0$. Furthermore, we write $(p, v) \vdash_V^* (p', v')$ if there is a sequence $(p_1, v_1) \vdash_V \dots \vdash_V (p_n, v_n)$ for some $n \geq 1$ such that $(p, v) = (p_1, v_1)$ and $(p', v') = (p_n, v_n)$. The reachability problem for \mathbb{Z} -VASS $_k^{\text{nz}}$ is defined as follows: given a \mathbb{Z} -VASS $_k^{\text{nz}}$ V and two configurations $(p, v), (p', v')$, does $(p, v) \vdash_V^* (p', v')$ hold? As shown in [15], this problem is NP-complete² for any fixed k .

LEMMA 6.10. *Let \mathcal{A}_1 be a deterministic strong reset PA and let \mathcal{A}_2 be a reach-reg. PA. The question $SR_\omega(\mathcal{A}_1) \cap RR_\omega(\mathcal{A}_2) = \emptyset$ is coNP-complete.*

PROOF. Let $\mathcal{A}_1 = (Q_1, \Sigma, p_0, \Delta_1, F_1, C)$ be of dimension d_1 and $\mathcal{A}_2 = (Q_2, \Sigma, q_0, \Delta_2, F_2, D)$ of dimension d_2 . By [11, Theorem 9] we can write $RR_\omega(\mathcal{A}_2)$ as $\bigcup_i U_i V_i^\omega$, where the U_i are finite word PA languages, and V_i are regular languages (in fact, the number of sets in the union is polynomially bounded by $|Q_2|$ and we can compute automata for these languages in polynomial time).

The idea for the proof of the lemma is as follows. Suppose $SR_\omega(\mathcal{A}_1) \cap RR_\omega(\mathcal{A}_2) \neq \emptyset$ and let α be an infinite word accepted by both automata. We can write $\alpha = u\beta$ such that $u \in U_i$ for some i . Let $(p, v) \in Q_1 \times \mathbb{N}^{d_1}$ be the configuration of \mathcal{A}_1 after reading u (as \mathcal{A}_1 is deterministic, there is a unique partial run of \mathcal{A}_1 on u , and as $\alpha = u\beta \in SR_\omega(\mathcal{A}_1)$, every visit of an accepting state while reading u is with good counter values). Hence, if we can verify the existence of such a word u and compute (p, v) , we can use the algorithm in the previous lemma to test $SR(\mathcal{A}_1^{(p,v)}) \cap L_\omega(\mathcal{B}) \neq \emptyset$ where $\mathcal{A}_1^{(p,v)}$ is defined as \mathcal{A}_1 but with initial configuration (p, v) (such a PA can be obtained from \mathcal{A}_1 by adding a fresh initial state that inherits all transitions from p and adds v to them) and \mathcal{B} is a Büchi automaton for V_i^ω .

In fact, we guess the existence of two finite words u and v such that (1) there is an infinite word β with $u\beta \in SR_\omega(\mathcal{A}_1) \cap RR_\omega(\mathcal{A}_2)$ (2) $u \in U_i$ for some i and (3) the unique partial run of \mathcal{A}_1 is in an accepting state after reading uv , and every visit of an accepting state while reading uv is with good counter values. We verify (2) and (3) using the NP-algorithm for the reachability problem for \mathbb{Z} -VASS $_k^{\text{nz}}$, yielding a deterministic strong reset PA $\mathcal{A}_1^{(p,v)}$ for a configuration $(p, v) \in Q_1 \times \mathbb{N}^{d_1}$ and a Büchi automaton \mathcal{B} as described above, which we use to call the algorithm in the previous lemma. To be precise, we build a product \mathbb{Z} -VASS $_2^{\text{nz}}$ V with $d_1 + d_2$ many counters. This system works in two phases: in the first phase it tests the existence of u ensuring that every visit of an accepting state of \mathcal{A}_1 is with good counter values using a zero test on the

²We note that our definition of \mathbb{Z} -VASS $_k^{\text{nz}}$ differs slightly from the definition in [15], as we allow $E \subseteq Q \times \mathbb{Z}^d \times Z \times Q$ instead of $E \subseteq Q \times \{-1, 0, 1\}^d \times Z \times Q$ only. However, this difference does not change the mentioned complexity for the reachability problem [15], see also [1, Section A.1].

first d_1 counters. After finding such a u , the system transitions into the second phase in which the last d_2 counters are not modified any more, and we check the existence of a v as mentioned above. After this check all counter are 0.

Let us give some more details on how to verify that every visit of an accepting state implies good counter values. Let $C = C(b_1, P_1) \cup \dots \cup C(b_k, P_k)$ for some $k \geq 1$. For every accepting state $f \in F_1$ we insert k states, say $f^{(1)}, \dots, f^{(k)}$, and a copy of f itself. We connect f to $f^{(i)}$ with a transition with update $-b_i$ and no zero-test. Then, for every period vector $p_i \in P_i$, we insert a loop on $f^{(i)}$ with update $-p_i$. Finally, every outgoing transition of $f^{(i)}$ is equipped with a zero-test on the first d_1 counters. This construction allows us to test membership of the current counter values in C , while resetting the counters in parallel.

Let $D = C(c_1, R_1) \cup \dots \cup C(c_\ell, R_\ell)$ for some $\ell \geq 1$. Formally, we define $V = (Q, \{0, d_1, d_1 + d_2\}, E)$ where

$$Q = Q_1 \times Q_2 \times \{1, 2\} \cup \{f^{(i)} \mid f \in F_1, i \leq k\} \\ \cup \{c_{f,q}^{(i)}, d_{f,q}^{(j)}, ef, q \mid f \in F_1, q \in Q_2, i \leq k, j \leq \ell\}$$

and

$$E = \{((p, q, 1), u \cdot v, 0, (p', q', 1)) \mid \\ \exists a \in \Sigma : (p, a, u, p') \in \Delta_1, (q, a, v, q') \in \Delta_2, p \notin F_1\} \\ \cup \{((p, q, 2), u \cdot 0, 0, (p', q', 2)) \mid \\ \exists a \in \Sigma \exists v \in \mathbb{N}^{d_2} : (p, a, u, p') \in \Delta_1, \\ (q, a, v, q') \in \Delta_2, p \notin F_1\} \\ \cup \{((f, q, 1), -b_i \cdot 0, 0, (f^{(i)}, q)) \mid f \in F_1, q \in Q_2, i \leq k\} \\ \cup \{((f^{(i)}, q), -p_i \cdot 0, 0, (f^{(i)}, q)) \mid \\ f \in F_1, q \in Q_2, p_i \in P_i, i \leq k\} \\ \cup \{((f^{(i)}, q), u \cdot v, d_1, (p', q', 1)) \mid \\ \exists a \in \Sigma : (f, a, u, p') \in \Delta_1, (q, a, v, q') \in \Delta_2, \\ f \in F_1, i \leq k\} \\ \cup \{((p, g, 1), u \cdot 0, 0, (p', q', 2)) \mid \\ \exists a \in \Sigma \exists v \in \mathbb{N}^{d_2} : (p, a, u, p') \in \Delta_1, (g, a, v, q') \in \Delta_2, \\ p \notin F_1, g \in F_2\} \\ \cup \{((f^{(i)}, g, 1), u \cdot 0, d_1, (p', q', 2)) \mid \\ \exists a \in \Sigma \exists v \in \mathbb{N}^{d_2} : (f, a, u, p') \in \Delta_1, (g, a, v, q') \in \Delta_2, \\ f \in F_1, g \in F_2, i \leq k\} \\ \cup \{((f, q, 2), -b_i \cdot 0, 0, c_{f,q}^{(i)}) \mid f \in F_1, q \in Q_2, i \leq k\} \\ \cup \{c_{f,q}^{(i)}, -p_i \cdot 0, 0, c_{f,q}^{(i)} \mid f \in F_1, q \in Q_2, p_i \in P_i, i \leq k\} \\ \cup \{c_{f,q}^{(i)}, 0 \cdot -c_j, 0, d_{f,q}^{(j)} \mid f \in F_1, q \in Q_2, i \leq k, j \leq \ell\} \\ \cup \{d_{f,q}^{(j)}, 0 \cdot -r_j, 0, d_{f,q}^{(j)} \mid f \in F_1, q \in Q_2, r_j \in R_j, j \leq \ell\} \\ \cup \{d_{f,q}^{(j)}, 0, d_1 + d_2, ef, q \mid f \in F_1, q \in Q_2, j \leq \ell\}.$$

Observe that V is computable in polynomial time.

We are now ready to present an NP-algorithm deciding whether $SR_\omega(\mathcal{A}_1) \cap RR_\omega(\mathcal{A}_2) \neq \emptyset$.

- (1) Guess $f \in F_1$ and $q \in Q_2$.
- (2) Construct V as above and test $((p_0, q_0, 1), 0) \vdash_V^* (ef, q, 0)$.
- (3) Finally, use the algorithm in the previous lemma to test $SR_\omega(\mathcal{A}_1^{f,0}) \cap L_\omega(\mathcal{B}_q) \neq \emptyset$, where \mathcal{B}_q is the Büchi automaton obtained from \mathcal{A}_2 by forgetting all vectors (and dropping C_2), and setting q as its initial state.

If all tests succeed, we conclude that the intersection of $SR_\omega(\mathcal{A}_1)$ and $RR_\omega(\mathcal{A}_2)$ is not empty. \square

COROLLARY 6.11. *Universal safety model checking and universal PA model checking are decidable for deterministic limit PA and deterministic strong reset PA, deterministic safety PA and deterministic co-Büchi PA.*

Recall that the existential PA model checking problem can equivalently be expressed as an intersection emptiness problem. As (non-deterministic) reach PA are closed under intersection and have a decidable emptiness problem, the existential PA model checking problem is decidable for them and hence also for their deterministic variants. This is also the case for reach-reg. PA: they are equivalent to limit PA [11], and we can show that limit PA are closed under intersection using the same argument as in the deterministic setting (see Lemma 5.2).

To show that the existential PA model checking problem is decidable (even for non-deterministic) Büchi PA, we use a recent result essentially stating that Ramsey-quantifiers in Presburger formulas can be eliminated in polynomial time [3]. Furthermore, the authors show how to use the Ramsey-quantifier to check liveness properties for systems with counters. In particular, the existence of an accepting run of a Büchi PA (answering the question whether the accepted ω -language is non-empty) can be expressed with a Presburger formula with a Ramsey-quantifier. Hence, checking if the intersection of the two ω -languages recognized by two Büchi PA can be tested by intersecting two Presburger-formulas and moving the quantifiers to the front. We refer to sections 4.1 and 8.2 in [3] for more information.

COROLLARY 6.12. *Existential PA model checking is decidable for (deterministic) reach PA, (deterministic) reach-reg. PA, (deterministic) limit PA, and (deterministic) Büchi PA.*

We show that deciding whether the intersection of two ω -languages recognized by deterministic strong reset PA is empty is undecidable. The result relies on the fact that the intersection of two such languages can encode non-terminating computations of two-counter machines.

A two-counter machine \mathcal{M} is a finite sequence of instructions

$$(1 : l_1)(2 : l_2) \dots (k-1 : l_{k-1})(k : \text{STOP})$$

where the first component of a pair (ℓ, l_ℓ) is the line number, and the second component is the instruction in line ℓ . An instruction is one of the following forms:

- $\text{Inc}(Z_i)$, where $i = 0$ or $i = 1$.
- $\text{Dec}(Z_i)$, where $i = 0$ or $i = 1$.
- $\text{If } Z_i = 0 \text{ goto } \ell' \text{ else } \ell''$, where $i = 0$ or $i = 1$, and $\ell', \ell'' \leq k$.

Instructions of the first or second form are called increments resp. decrements, while instructions of the latter form are called zero-tests. A configuration of \mathcal{M} is a tuple $c = (\ell, z_0, z_1)$, where $\ell \leq k$ is the current line number, and $z_0, z_1 \in \mathbb{N}$ are the current counter values of Z_0 and Z_1 respectively. We say c *derives* into its unique successor configuration c' , written $c \vdash c'$, as follows.

- If $l_\ell = \text{Inc}(Z_0)$, then $c' = (\ell + 1, z_0 + 1, z_1)$.
- If $l_\ell = \text{Inc}(Z_1)$, then $c' = (\ell + 1, z_0, z_1 + 1)$.
- If $l_\ell = \text{Dec}(Z_0)$, then $c' = (\ell + 1, \max\{z_0 - 1, 0\}, z_1)$.
- If $l_\ell = \text{Dec}(Z_1)$, then $c' = (\ell + 1, z_0, \max\{z_1 - 1, 0\})$.
- If $l_\ell = \text{If } Z_0 = 0 \text{ goto } \ell' \text{ else } \ell''$, then $c' = (\ell', z_0, z_1)$ if $z_0 = 0$, and $c' = (\ell'', z_0, z_1)$ if $z_0 > 0$.
- If $l_\ell = \text{If } Z_1 = 0 \text{ goto } \ell' \text{ else } \ell''$, then $c' = (\ell', z_0, z_1)$ if $z_1 = 0$, and $c' = (\ell'', z_0, z_1)$ if $z_1 > 0$.
- If $l_\ell = \text{STOP}$, then c has no successor configuration.

The unique computation of \mathcal{M} is a finite or infinite sequence of configurations $c_0 c_1 c_2 \dots$ such that $c_0 = (1, 0, 0)$ and $c_i \vdash c_{i+1}$ for all $i \geq 0$. Observe that the computation is finite if and only if the instruction $(k : \text{STOP})$ is reached. If this is the case, we say \mathcal{M} terminates. Given a two-counter machine \mathcal{M} , it is undecidable to decide whether \mathcal{M} terminates or not [21].

In the following we assume w.l.o.g. that our two-counter machines satisfy the guarded-decrement property [12] which guarantees that every decrement does indeed change a counter value: every decrement $(\ell : \text{Dec}(Z_i))$ is preceded by a zero-test of the form $(\ell - 1, \text{If } Z_i = 0 \text{ goto } \ell + 1 \text{ else } \ell)$. Note that this modification does not change the termination behavior of a two-counter machine, as decrementing a counter whose value is already zero does not have an effect.

LEMMA 6.13. *The intersection emptiness problem for deterministic strong reset PA is undecidable.*

PROOF. We can encode infinite computations of two-counter machines as infinite words over $\Sigma = \{a, b, 1, 2, \dots, k\} \cup \Sigma_1$, where $\Sigma_1 = \{I_a, I_b, D_a, D_b, Z_a, Z_b, \bar{Z}_a, \bar{Z}_b\}$. The idea is as follows. Let $c = (\ell, z_0, z_1)$ be a configuration of \mathcal{M} . We encode c as a finite word $w_c = \ell ux \in \Sigma^*$, where $\ell \in \{1, 2, \dots, k\}$ encodes the current line number, $u \in \{a, b\}^*$ with $|u|_a = z_0$ and $|u|_b = z_1$ encodes the current counter values, and $x \in \Sigma_1$ encodes the instruction l_ℓ of line ℓ as follows:

- If $l_\ell = \text{Inc}(Z_0)$, then $x = I_a$, and if $l_\ell = \text{Inc}(Z_1)$, then $x = I_b$.
- If $l_\ell = \text{Dec}(Z_0)$, then $x = D_a$, and if $l_\ell = \text{Dec}(Z_1)$, then $x = D_b$.
- If $l_\ell = \text{If } Z_0 = 0 \text{ goto } \ell' \text{ else } \ell''$, and the line number of the unique successor configuration of c is ℓ' , then $x = Z_a$ (that is, the zero-test is successful). Analogously with $x = Z_b$.
- If $l_\ell = \text{If } Z_0 = 0 \text{ goto } \ell' \text{ else } \ell''$, and the line number of the unique successor configuration of c is ℓ'' , then $x = \bar{Z}_a$ (that is, the zero-test fails). Analogously with $x = \bar{Z}_b$.

Let $w_c, w_{c'} \in \Sigma^*$ be two words encoding two configurations of \mathcal{M} . We call $w_c \cdot w_{c'}$ *correct* if $c \vdash c'$. Hence, we can encode a unique infinite computations $c_0 c_1 c_2 \dots$ as an infinite word $w_{c_0} w_{c_1} w_{c_2} \dots$. We show that the ω -language $L = \{w_{c_0} w_{c_1} w_{c_2} \dots\}$ can be written as the intersection of two deterministic strong reset PA ω -languages.

Let

$$L_1 = \{w_{c_0} w_{c_1} w_{c_2} \dots \mid w_{c_{2i}} w_{c_{2i+1}} \text{ is correct for every } i \geq 0\}, \text{ and}$$

$$L_2 = \{w_{c_0} w_{c_1} w_{c_2} \dots \mid w_{c_{2i+1}} w_{c_{2i+2}} \text{ is correct for every } i \geq 0\}.$$

Observe that $L_1 \cap L_2 = L$, and L is empty if and only if the unique computation of \mathcal{M} terminates. Hence, it remains to show that L_1 and L_2 are recognized by deterministic strong reset PA. We argue for L_1 ; the argument for L_2 is very similar. The idea is as follows: We construct a deterministic strong reset PA \mathcal{A}_1 with five counters that tests the correctness of two consecutive encodings of configurations, say $w_{c_{2i}} \cdot w_{c_{2i+1}} = \ell_1 u_1 x_1 \cdot \ell_2 u_2 x_2$ with $\ell_1, \ell_2 \in \{1, 2, \dots, k\}$, $u_1 u_2 \in \{a, b\}^*$ and $x_1, x_2 \in \Sigma_1$. First observe that checking whether ℓ_2 is indeed the correct line number (that is, the correct successor of ℓ_1) can be hard-coded into the state space of \mathcal{A}_1 : if x_1 encodes an increment or decrement, we expect $\ell_2 = \ell_1 + 1$, and if x_1 encodes a successful or failing zero-test $\text{If } Z_i = 0 \text{ goto } \ell' \text{ else } \ell''$, we expect $\ell_2 = \ell'$ or $\ell_2 = \ell''$, respectively. Four counters of \mathcal{A}_1 are used to count the numbers of a 's and b 's in u_1 and u_2 , respectively. Then, if $x = I_a$, we expect $|u_2|_a = |u_1|_a + 1$ and $|u_2|_b = |u_1|_b$, and so on. To be able to perform the correct check, we also encode x_1 into the state space as well as the fifth counter by counting modulo $|\Sigma_1|$. Observe that the guarded-decrement property ensures that decrements are handled correctly. Hence, \mathcal{A}_1 has two sets of states counting the numbers of a 's and b 's of u_1 and u_2 , accordingly, as well as a set of accepting states that is used to check the counter values.

After such a check, \mathcal{A}_1 resets, and continues with the next two (encodings of) configurations. The automaton for L_2 works in the same way, but skips the first configuration. \square

As shown, the intersection emptiness problem for deterministic strong reset PA is undecidable, implying the same for deterministic weak reset PA, and hence their existential PA model checking problems. Recall that this problem is also undecidable for deterministic safety PA and deterministic co-Büchi PA.

COROLLARY 6.14. *Existential PA model checking for deterministic strong reset PA, deterministic weak reset PA, deterministic safety PA and deterministic co-Büchi PA is undecidable.*

7 CONCLUSION

We have studied the deterministic variants of weak reset PA, strong reset PA, reach-reg. PA, and most notably limit PA. The latter are closed under the Boolean operations and hence all common decision problems are decidable for them, including the classical model checking problems. Closely related to model checking problems are synthesis problems. Here, the problem is to generate a model from a system specification (which is correct by construction). Gale-Stewart games play a key role in solving such synthesis problems [9]. However, these games are undecidable when winning conditions are specified by automata whose emptiness or universality problem is undecidable. Our decidability results for deterministic limit PA raise the interesting and important question whether Gale-Stewart games can be solved when their winning condition is expressed by these automata.

In future work we further plan to study the regular separability problem for these models, that is, given two ω -languages L_1, L_2 recognized by PA operating on infinite words, is there an ω -regular

language L with $L_1 \subseteq L$ and $L_2 \cap L = \emptyset$. Solving this problem can be used as an alternative approach to solving existential PA model checking, as (regular) separability implies intersection emptiness. It has already been studied for some related models, e.g. PA on finite words [5] and Büchi VASS [2]. Furthermore, it remains to classify the intersections of all incomparable models and thereby to provide a fine grained “map of the universe” for Parikh recognizable ω -languages.

REFERENCES

- [1] Pascal Baumann, Flavio D’Alessandro, Moses Ganardi, Oscar Ibarra, Ian McQuillan, Lia Schütze, and Georg Zetsche. 2023. Unboundedness Problems for Machines with Reversal-Bounded Counters. In *Foundations of Software Science and Computation Structures*. Springer Nature Switzerland, Cham, 240–264.
- [2] Pascal Baumann, Roland Meyer, and Georg Zetsche. 2023. Regular Separability in Büchi VASS. arXiv:2301.11242 [cs.FL]
- [3] Pascal Bergsträßer, Moses Ganardi, Anthony W. Lin, and Georg Zetsche. 2024. Ramsey Quantifiers in Linear Arithmetics. *Proc. ACM Program. Lang.* 8, POPL (2024). <https://doi.org/10.1145/3632843>
- [4] Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem. 2018. *Handbook of Model Checking* (1st ed.). Springer Publishing Company, Incorporated.
- [5] Lorenzo Clemente, Wojciech Czerwiński, Sławomir Lasota, and Charles Paperman. 2016. Regular Separability of Parikh Automata. arXiv:1612.06233 [cs.FL]
- [6] Leonard Eugene Dickson. 1913. Finiteness of the Odd Perfect and Primitive Abundant Numbers with n Distinct Prime Factors. *American Journal of Mathematics* 35, 4 (1913), 413–422.
- [7] Henning Fernau and Ralf Stiebe. 2008. Blind Counter Automata on omega-Words. *Fundam. Inform.* 83 (2008), 51–64.
- [8] Diego Figueira and Leonid Libkin. 2015. Path Logics for Querying Graphs: Combining Expressiveness and Efficiency. In *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (LICS ’15)*. IEEE, 329–340.
- [9] David Gale and F. M. Stewart. 1953. *Infinite Games with Perfect Information*. Princeton University Press, Princeton, 245–266. <https://doi.org/doi:10.1515/9781400881970-014>
- [10] Sheila A. Greibach. 1978. Remarks on blind and partially blind one-way multi-counter machines. *Theoretical Computer Science* 7, 3 (1978), 311–324.
- [11] Mario Grobler, Leif Sabellek, and Sebastian Siebertz. 2023. Remarks on Parikh-recognizable omega-languages.
- [12] Shibashis Guha, Ismaël Jecker, Karoliina Lehtinen, and Martin Zimmermann. 2022. Parikh Automata over Infinite Words. In *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2022) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 250)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 40:1–40:20.
- [13] Christoph Haase. 2018. A Survival Guide to Presburger Arithmetic. *ACM SIGLOG News* 5, 3 (2018), 67–82.
- [14] Christoph Haase and Simon Halfon. 2014. *Integer Vector Addition Systems with States*. Springer International Publishing, 112–124. https://doi.org/10.1007/978-3-319-11439-2_9
- [15] Christoph Haase and Georg Zetsche. 2021. Presburger arithmetic with stars, rational subsets of graph groups, and nested zero tests. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (Vancouver, Canada) (LICS ’19)*. IEEE Press, Article 58, 14 pages.
- [16] Luisa Herrmann, Vincent Peth, and Sebastian Rudolph. 2023. Decidable (Ac)counting with Parikh and Muller: Adding Presburger Arithmetic to Monadic Second-Order Logic over Tree-Interpretable Structures. arXiv:2305.01962 [cs.LO]
- [17] Oscar H. Ibarra. 1978. Reversal-Bounded Multicounter Machines and Their Decision Problems. *J. ACM* 25, 1 (1978), 116–133.
- [18] Felix Klaedtke and Harald Rueß. 2003. Monadic Second-Order Logics with Cardinalities. In *Automata, Languages and Programming*. Springer, Berlin, Heidelberg, 681–696.
- [19] Lawrence H. Landweber. 1969. Decision problems for ω -automata. *Mathematical systems theory* 3 (1969), 376–384.
- [20] Robert McNaughton. 1966. Testing and generating infinite sequences by a finite automaton. *Information and control* 9, 5 (1966), 521–530.
- [21] Marvin L. Minsky. 1967. *Computation: Finite and Infinite Machines*. Prentice-Hall.
- [22] Wolfgang Thomas. 1991. *Automata on Infinite Objects*. MIT Press, Cambridge, MA, USA, 133–191.
- [23] Wolfgang Thomas et al. 2002. *Automata, logics, and infinite games: a guide to current research*. Vol. 2500. Springer Science & Business Media.